09:37:02 OCA PAD AMENDMENT - PROJECT HEADER INFORMATION 11/17/95

Active

Project #: E-15-623 Cost share #: Rev #: 3
Center # : 10/24-6-R8390-0A1 Center shr #: OCA file #:
Work type : RES
Contract#: OSP-94-11-668-001 Mod #: ADMIN Document : SUBCONT
Prime #: DAAE07-94-C-R062 Contract entity: GTRC

Subprojects ? : N CFDA: NA
Main project #: PE #: NA

Project unit: OIP Unit code: 03.010.200
Project director(s):
GOURISANKAR V ENGR COLL (Q04)894-7772

Sponsor/division names: CLARK ATLANTA UNIVERSITY / ATLANTA, GA
Sponsor/division codes: 400 / 094

Award period: 941212 to 951130 (performance) 960131 (reports)

Sponsor amount	New this change	Total to date
Contract value	0.00	92,145.00
Funded	0.00	92,145.00
Cost sharing amount		0.00

Does subcontracting plan apply ?: N

Title: PRODUCE DATA CONVERSION FACILITY FOR TACOM

PROJECT ADMINISTRATION DATA

OCA contact: Ina R. Lashley 894-4820

Sponsor technical contact

DR. O. OLATIDOYE
(404)880-6940

CLARK ATLANTA UNIVERSITY
223 JAMES P. BRWLEY DRIVE, S.W.
ATLANTA, GA 30314

Sponsor issuing office

MS. CAROL E JOHNSON
(404)880-6985

DIRECTOR, PROJECT INITIATION
CLARK ATLANTA UNIVERSITY
223 JAMES P. BRAWLEY DRIVE, S.W.
ATLANTA, GA 30314

Security class (U,C,S,TS) : U ONR resident rep. is ACO (Y/N): N
Defense priority rating : NA NA supplemental sheet
Equipment title vests with: Sponsor X GIT
NONE PROPOSED.
Administrative comments -
ADMINISTRATIVE MODIFICATION ISSUED TO CHANGE PROJECT NUMBER FROM B-10-F78 TO E-15-623

GEORGIA INSTITUTE OF TECHNOLOGY
OFFICE OF CONTRACT ADMINISTRATION

NOTICE OF PROJECT CLOSEOUT

SR314

Closeout Notice Date 01/12/96

Project No. E-15-623 Center No. 10/24-6-R8390-0A1

Project Director GOURISANKAR V School/Lab OIP

Sponsor CLARK ATLANTA UNIVERSITY/ATLANTA, GA

Contract/Grant No. OSP-94-11-668-001 Contract Entity GTRC

Prime Contract No. DAAE07-94-C-R062

Title PRODUCE DATA CONVERSION FACILITY FOR TACOM

Effective Completion Date 951130 (Performance) 960131 (Reports)

Closeout Actions Required:	Y/N	Date Submitted
Final Invoice or Copy of Final Invoice	Y	
Final Report of Inventions and/or Subcontracts	Y	
Government Property Inventory & Related Certificate	Y	
Classified Material Certificate	N	
Release and Assignment	Y	
Other	N	

Comments

Subproject Under Main Project No.

Continues Project No.

Distribution Required:

Project Director	Y
Administrative Network Representative	Y
GTRI Accounting/Grants and Contracts	Y
Procurement/Supply Services	Y
Research Property Managment	Y
Research Security Services	N
Reports Coordinator (OCA)	Y
GTRC	Y
Project File	Y
Other	N
	N

NOTE: Final Patent Questionnaire sent to PDPI.

E-15-623

2, 3

PRODUCT DATA CONVERSION FACILITY

Pro/ENGINEER - I/EMS

Prime Contract No. DAAE07-94-C-R062
Sub Contract No. OSP-94-11-668-001
Final Report
November 30, 1995

Prepared by
Center for Information Technology Insertion
College of Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332-0140
Phone: (404) 894-7772
FAX: (404) 894-9812

TABLE OF CONTENTS

1.0 INTRODUCTION 1

2.0 SCOPE OF THE CONVERSION PROCESS 1

3.0 ASSEMBLY CONVERSION PROCESS OVERVIEW..... 2
3.1 DETAILED DISCUSSION OF CONVERSION FROM EMS TO PROENGINEER..... 3
3.2 DETAILED DISCUSSION OF CONVERSION FROM PROENGINEER TO EMS..... 4

4.0 FUNCTIONAL BREAKDOWN OF CONVERSION SOFTWARE MODULES..... 6

5.0 CONVERSION FILE SYSTEM STRUCTURE..... 6

6.0 CONCLUDING REMARKS..... 8

APPENDIX A: LIST OF UTILITIES DEVELOPED..... 9

APPENDIX B: USER GUIDE FOR THE ASSEMBLY CONVERSION PROCESS 18

1.0 INTRODUCTION

The objective of the conversion facility developed in this project is to exchange assemblies of 3-D geometric models between Intergraph's EMS version 3.0 and Parametric Technology's ProEngineer Release 14 using IGES. The basic components of the assembly are 3-D geometric models, which are usually referred to as "parts" in ProEngineer and "designs" in EMS. The assemblies are created from these components in a hierachical manner. The assembly models contain references to the component models which are oriented as necessary during the assembly process.

Though both ProEngineer and EMS provide the capability to export assembly models into IGES, neither has the built-in capability to import an IGES file representing an assembly and recreate the assembly model. Further, EMS export of assembly models into IGES does not retain the assembly structure in the IGES file at all. In the case of ProEngineer, though the assembly structure is retained in the IGES file, the IGES file so generated cannot be imported into EMS or even looped back into ProEngineer. The conversion facility that has been developed overcomes this limitation of the built-in IGES import/export capabilities of EMS and ProEngineer. The conversion facility preserves the hierarchical structure contained in the assembly in the source system during the entire conversion process.

2.0 SCOPE OF THE CONVERSION PROCESS

The tools and procedures developed for the conversion of assemblies is quite different from that developed for the conversion of stand-alone models that have no external references. The primary emphasis in this case is to keep the hierarchical structure of the assembly intact from the source to the target CAD system.

At present, the two CAD systems lack the ability to exchange CSG elements through IGES. Therefore, the solid models in the source CAD system are translated into IGES as precise surface representations. After conversion, the surface representation of the components can be changed into solid models using the built-in capabilities of the CAD system. Further, additional "intelligence" present in the source CAD model, such as the "parametric" design in ProEngineer or "variational" geometric design in EMS are not preserved during the conversion since the exchange is through IGES.

The conversion process has been designed around a baseline geometry and assembly creation process. Departures from this baseline process may adversely affect the results of conversion. The constraints placed on the creation process are meant to improve the "transferability" of assemblies and components.

In view of this, the assemblies are restricted to one level of nesting in the assembly file (a total of two levels of decompostion). If a real assembly has more than two levels of decomposition, it should be broken down into simpler assemblies in preparation for translation. There is no restriction on the number of components that can be assembled at a given level of the assembly.

Each of the lowest level components of the assembly should be created as a separate "design" or "part" file containing a solid in the native CAD system. The assembly models should then be created that contain only references to the component model files. No geometry is to be created or modified directly in the assembly model.

Care should be taken while naming models that are to be converted. The names used should not violate the naming conventions of either the source or the target CAD system. For example, EMS does not accept filenames with a "-" (hyphen) or filenames longer than 14 characters in length. Similarly, ProEngineer does not accept filenames with an embedded "." (period). Further, while naming the models in EMS it is necessary to take care that the extensions used in naming do not end in a numeral. This restriction is placed because EMS adds a numeral to the extension while creating unique names for reference file attachment names and the conversion procedure uses the reference file attachment names in creating names for IGES files.

3.0 ASSEMBLY CONVERSION PROCESS OVERVIEW

The conversion process developed for exchanging assembly models between EMS and ProEngineer involves three major stages - export, flavoring and import. These stages are as shown in Figure 1: Conversion Process Overview.

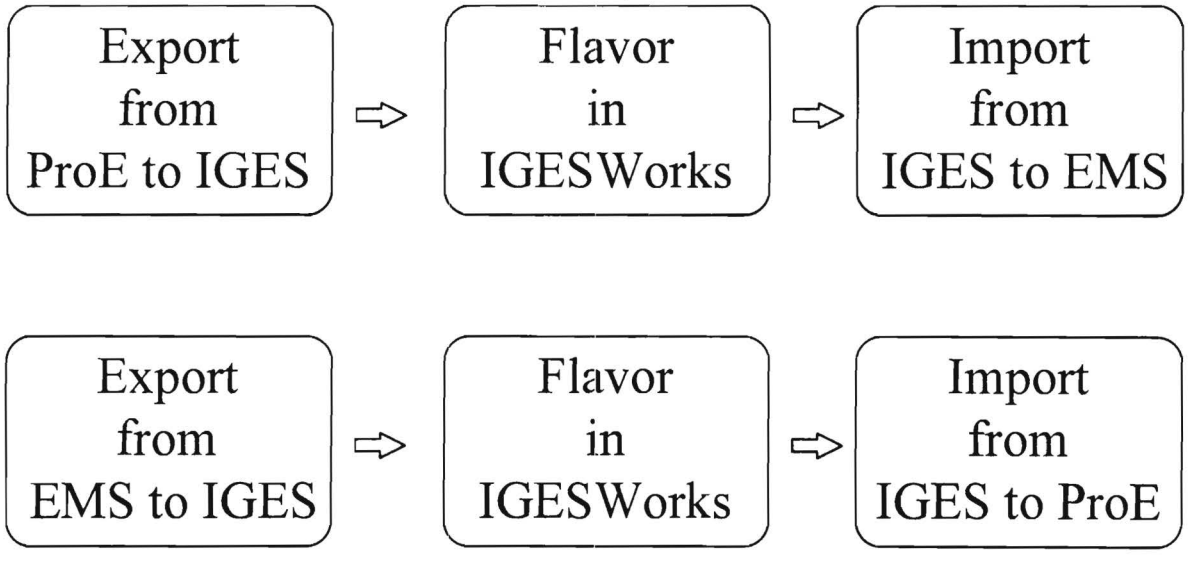

Figure 1: Conversion Process Overview

The first stage is concerned with the export of the assembly model from the source CAD system into IGES. The second stage, involves modifying ("flavoring") the IGES file from the source system in preparation for import into the target CAD system. The third stage is designed to import the IGES file(s) into the target CAD system and to re-create the assembly there. The first and third stages consist of operations carried out mainly in the source and target CAD systems respectively, while the second stage operations take place mainly in the IGESWorks

environment. IGESWorks (from International TechneGroup Incorporated) is a general purpose IGES file viewing and flavoring software that provides tools to extract information from and to modify IGES files.

The specific actions taken during each of the stages of conversion depends on the source and target CAD system. Therefore the details of conversion from EMS to ProEngineer and vice versa are discussed separately in the following sections.

3.1 DETAILED DISCUSSION OF CONVERSION FROM EMS TO PROENGINEER

Figure 2: EMS to ProEngineer Conversion Flow, depicts the steps involved in the conversion of an EMS assembly model into ProEngineer through IGES.

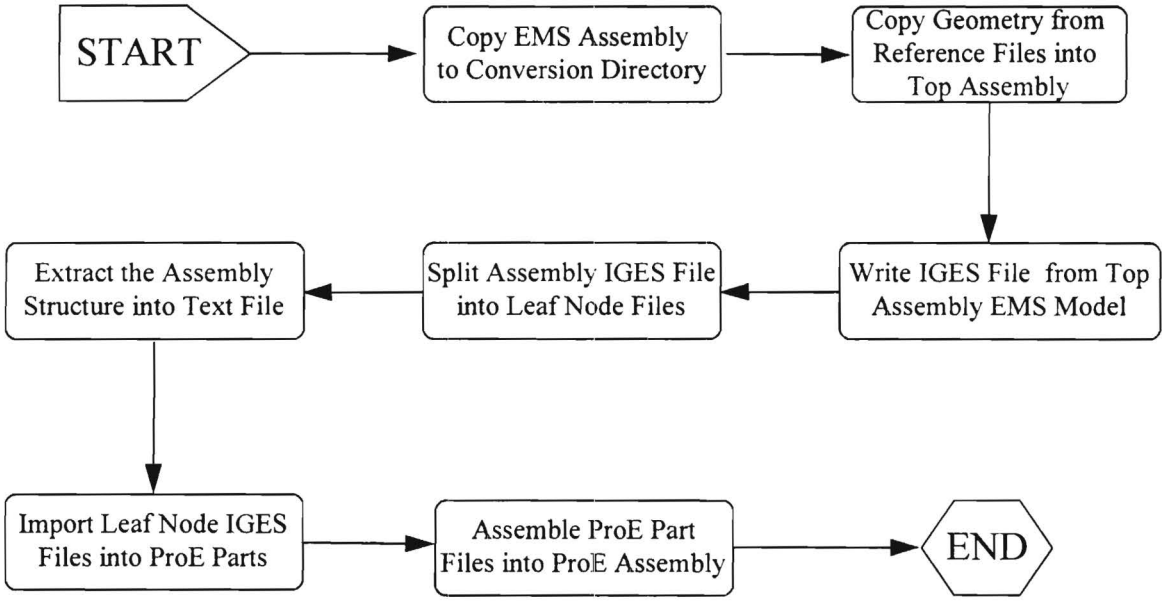

Figure 2: EMS to ProEngineer Conversion Flow

In order for all the utilities involved in the conversion process to work together in a seamless fashion it is necessary to organize the directory/file structure for all the utilities and the data files. This organization is explained in full in a later section, titled, "Conversion File System Structure" Further, it is essential that the stepwise instructions as described in Appendix B be followed exactly.

The first step is to copy the EMS assembly files to the proper location in the Conversion Directory structure. A utility called "copyasmems" is provided to perform this copying. It is important that the operating system commands (cp or ln) not be used for this purpose. Doing so would disrupt the internal ID's maintained in the EMS assembly to relate the reference files to the assembly files.

Before exporting the EMS assembly to IGES it is necessary to pre-process the assembly model within EMS. An IGES file written directly from the EMS assembly without this pre-processing lacks the information inherent in the structure of the assembly.

During pre-processing, all the elements in the component files are grouped together into a single EMS "graphic group" and then copied as a unit into the assembly file that references it. It is necessary to copy each instance of the reference file (component) with its proper orientation into the assembly file. This is accomplished using Parametric Programming Language (or PPL - the scripting language for EMS) programs under the control of the Unix script, "wrigasmems". During this process a certain level of user interaction with the operation of the PPL program is necessary in order to complete the copying of the component geometry into the assembly file. At the end of this pre-processing the copy of the EMS assembly in the Conversion Directory differs from the original assembly model, in that, all geometry from the original reference files have been copied directly into the assembly model itself and all the reference files have been detached. Also, the geometry from the individual components of the assembly have been organized into "graphic group" elements that mirror the hierarchical structure of the original EMS assembly. This EMS model is then exported to IGES using EMS' built-in I/IGES processor.

The IGES file that results retains the hierarchical structure of the EMS assembly through nested IGES group entities (#402). This IGES file needs further flavoring before being imported into ProEngineer. This is done using IGESWorks scripts.

The IGESWorks script, "splitp402.scr" is employed to split the IGES file from EMS into IGES files that represent the "leaf nodes" of the assembly tree. These contain geometry properly oriented for origin to origin assembly in ProEngineer. Also, the assembly structure as represented in the IGES file is extracted into text files for use later by the utility that recreates the assembly in ProEngineer.

The next stage of conversion takes place in the environment of the target system, ProEngineer, under the control of the Unix utility, "rdigasmproe". The "leaf node" IGES files are first read into ProEngineer part files using ProEngineer's built-in IGES import utility, "proigsutil". The ProEngineer part files are then assembled origin to origin to form the ProEngineer assembly model. A ProEngineer "trail file" is run to carry out the assembly. The text files that were created earlier in the conversion process describing the assembly structure in EMS are utilitized in recreating the proper assembly structure in ProEngineer.

3.2 DETAILED DISCUSSION OF CONVERSION FROM PROENGINEER TO EMS

Figure 3: ProEngineer to EMS Conversion Flow, depicts the steps involved in the conversion of a ProEngineer assembly model into EMS through IGES.

The first step is to copy the ProEngineer assembly files to the proper location in the Conversion Directory structure. A utility called "copyasmproe" is provided to perform this copying.

Though, unlike with EMS, it is alright to use the operating system commands (cp or ln) for this purpose, it is safer and more convenient to use this utility.

No pre-processing is necessary in ProEngineer before an IGES file is exported from the assembly model. The next step in conversion is then to write to IGES from the ProEngineer assembly model using ProEngineer's "pro_batch" utility. This creates an IGES file representing the top-level of the ProEngineer assembly which contains references (IGES entity type #416) to other IGES files representing the components of this assembly. The references are nested in the same manner as the assembly in ProEngineer. This collection of IGES files cannot be imported directly into EMS and needs further flavoring in IGES/Works.

The first flavoring action is to incorporate all the geometry in the reference IGES files into the main assembly IGES file. This is accomplished using the IGES/Works script, "explode.scr" which creates a single IGES file containing IGES entities of type #408 (subfigure instances) that point to entities of type #308 (subfigure definitions). Since EMS 3.0 IGES import does not correctly interpret the IGES #408/#308 combination, it is necessary to create individual IGES files for each leaf node of the assembly in the correct orientation for assembly. The IGES/Works script that performs this flavoring action is named "tm408fix.scr". Further, the assembly structure represented in the IGES files is extracted and stored in text files using another script, "getasmstr.scr".

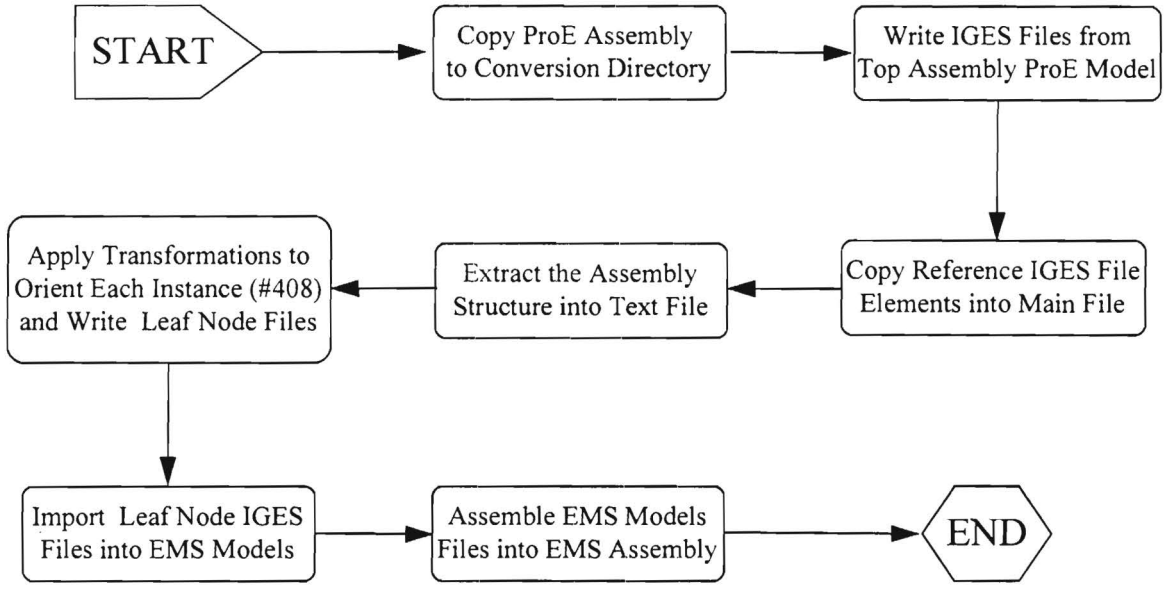

Figure 3: ProEngineer to EMS Conversion Flow

The final stage of the ProEngineer to EMS assembly conversion takes place in the EMS environment. The leaf node IGES files are imported into EMS design files using EMS' built-in IGES read utility, I/IGES. These design files are, in turn, assembled into an EMS assembly model using the PPL script, "assembl.u". Reading the IGES files and their assembly take place under the control of a single Unix script, "rdigasmems".

4.0 FUNCTIONAL BREAKDOWN OF CONVERSION SOFTWARE MODULES

The software utilities developed to perform the conversion of assembly models between EMS and ProEngineer can be classified into two main categories - "EMS to ProEngineer Facility" and "ProEngineer to EMS Facility". Each of these main categories can be subdivided further into a "Write IGES Facility" and a "Read IGES Facility". This functional breakdown of the software modules is shown in Figure 4: Functional Breakdown of Conversion Modules. All flavoring actions that are required are performed as part of the "Write IGES" process and take place in the IGES/Works environment.

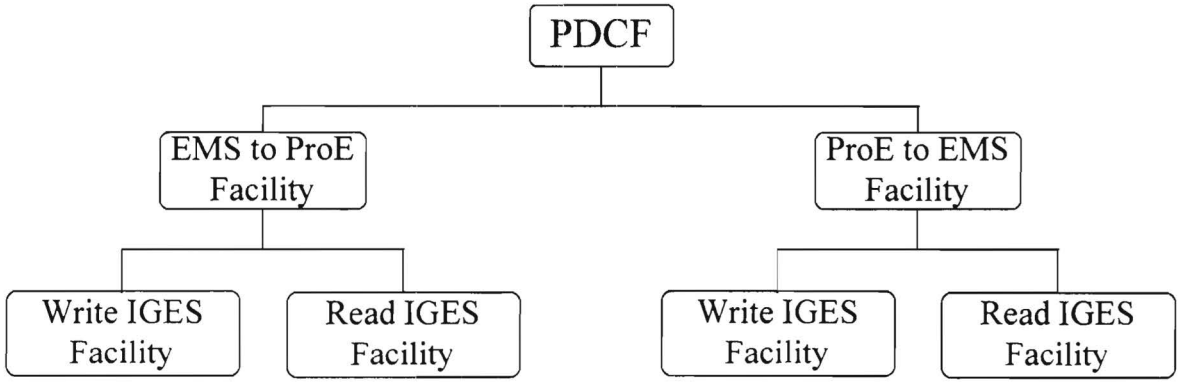

Figure 4: Functional Breakdown of Conversion Modules

5.0 CONVERSION FILE SYSTEM STRUCTURE

For the seamless operation of the utilities involved in the conversion it is necessary to organize the directory/file structure for all the utilities and the data files. Figure 5: Conversion File Structure, depicts this organization. The home directory of the conversion account (pdcf) contains the directory named "convert" which has subdirectories named "ems" and "proe". The utilities developed by the Georgia Tech team are placed in the "scripts" subdirectory under

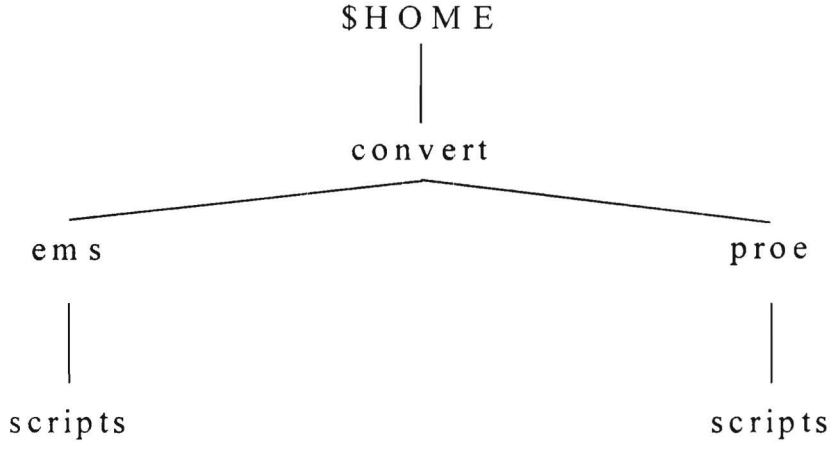

Figure 5: Conversion File Structure

"ems" and "proe" directories.

The utilities that deal with the EMS end of the conversion process are stored in the "$HOME/convert/ems/scripts" directory . These comprise the "Write IGES Facility" module for EMS to ProEngineer conversion as well as the "Read IGES Facility" module for ProEngineer to EMS conversion. Similarly, the utilities that deal with the ProEngineer end of the conversion process are stored in the "$HOME/convert/proe/scripts" directory . These comprise the "Write IGES Facility" module for ProEngineer to EMS conversion as well as the "Read IGES Facility" module for EMS to ProEngineer conversion. These two directories must be included in the PATH environment variable of the pdcf account. Also, the configuration file, $HOME/.cisetup should include the directory, "$HOME/convert/ems/scripts" in the search path for PPL program files. A sample ".cisetup" configuration file is stored in the directory, "$HOME/convert/ems/scripts". The configuration file used by the "Write IGES Facility" module for ProEngineer is the "config.pro" file in "$HOME/convert/proe/scripts".

The organization of data files during the conversion process from EMS to ProEngineer is shown in Figure 6: Conversion Data Organization - EMS to ProEngineer. The source EMS assembly and all its components are first copied into the directory, "assemwrite" in "$HOME/convert/ems" using the utility, "copyasmems". As conversion proceeds, the directories "flavor", "refdata", "childata", "iges", "log" and "proe" are created within the directory, "assemwrite". The directory, "flavor" holds the IGES files while they are being flavored using IGES/Works scripts. The directories "refdata" and "childata" contain files holding information on the assembly structure represented in the original EMS model and in the IGES files, respectively. The IGES files that are produced at the end of the flavoring actions in IGES/Works are held in the "iges" directory. The log files from the IGES import, export and flavoring actions are stored in the "log" directory. Finally, the ProEngineer part and assembly files that are created at the end of the conversion process are stored in the directory, "proe".

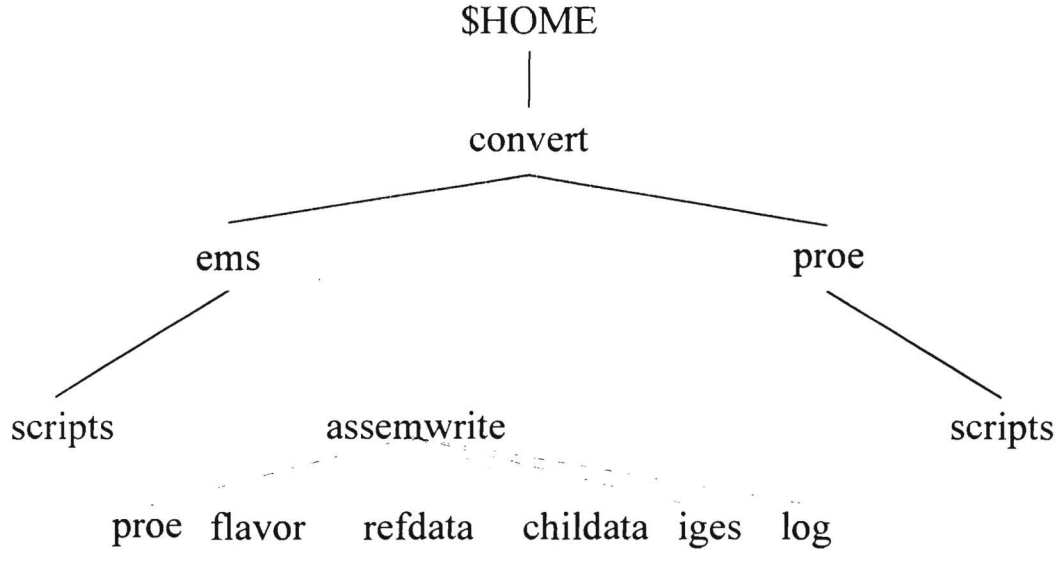

Figure 6: Conversion Data Organization - EMS to ProEngineer

The organization of data files during the conversion process from ProEngineer to EMS is shown in Figure 7: Conversion Data Organization - ProEngineer to EMS. The source ProEngineer assembly and all its components are first copied into the directory, "assemwrite" in "$HOME/convert/proe" using the utility, "copyasmproe". As conversion proceeds, the directories "flavor", "childata", "iges", "log" and "ems" are created within the directory, "assemwrite". The directory, "flavor" holds the IGES files while they are being flavored using IGES/Works scripts. The directory "childata" contains files holding information on the assembly structure represented in the IGES files. The IGES files that are produced at the end of the flavoring actions in IGES/Works are held in the "iges" directory. The log files from the IGES import, export and flavoring actions are stored in the "log" directory. Finally, the EMS design files and assembly files that are created at the end of the conversion process are stored in the directory, "ems".

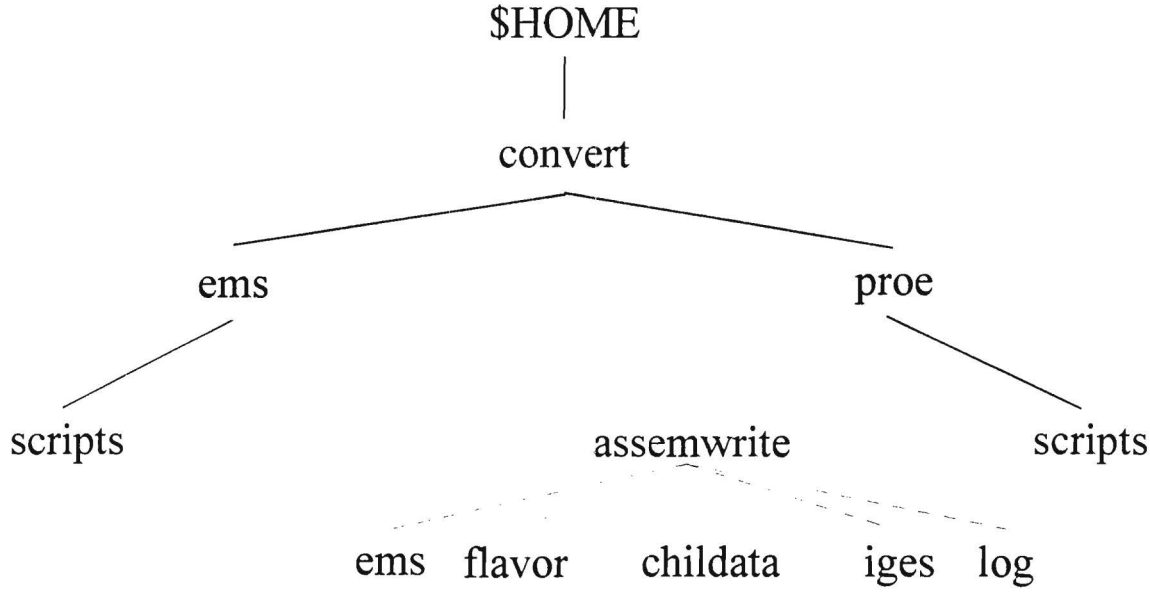

Figure 7: Conversion Data Organization - ProEngineer to EMS

6.0 CONCLUDING REMARKS

The techniques, procedures, and utilities developed during the course of this project provide a significant improvement in the ability to convert CAD assembly models between ProEngineer version 14 and EMS 3.0. The primary benefit of the conversion process described here is the ability to capture and retain the assembly hierarchy represented in the source CAD system throughout the entire process and to re-create it in the target CAD system. However, since the exchange of data is through IGES and the two CAD systems are, at present, not able to exchange Solid elements through IGES, only a surface representation of the models is available in the target system at the end of conversion. It is left to the user to exploit the capabilities of the target CAD system to re-create solid models from the surface models through operations such as "sewing up".

APPENDIX A

LIST OF UTILITIES DEVELOPED

The following is the list of utilities developed for the conversion of assembly models between EMS and ProEngineer. The list includes the usage instruction and a brief description of each utility.

1. Program: assembl.u (PPL script)

   Product: EMS PowerPak 03.00.00.34

   Author: Vasu Kaladi
   CITI, Georgia Tech
   vkaladi@cad.gatech.edu

   Usage: ci=assembl.u child_list_file

   Comments: This program creates an EMS assembly from input IGES file from ProEngineer after processing through flvigasmproe

2. Program: copyasmems (Unix script)

   Author: Vasu Kaladi
   CITI, Georgia Tech
   e-mail: vkaladi@cad.gatech.edu
   Usage: copyasmems top_assembly_name target_dir

   Comments: This utility takes as input the name of the top level EMS assembly model file name and the name of the target directory in $HOME/convert/ems to copy the source EMS assembly and all its reference files to.

3. Program: countref (Unix script)

   Author: Vasu Kaladi
   CITI, Georgia Tech
   e-mail: vkaladi@cad.gatech.edu

   Usage: countref reflist

   where reflist is the name of the reference list file to search.

Comments: Finds the number of reference file attachments listed in the reference list file.

4. Program: dot_uline (Unix script)

Author: Vasu Kaladi
CITI, Georgia Tech
e-mail: vkaladi@cad.gatech.edu

Usage: dot_uline name

where name is the input string where "." (dot)s are to be replaced with "_" (underline).

Comments: This is for use in the script, "splitc402.scr" to replace "." with "_" in the names of IGES files to be output to make them acceptable to ProEngineer..

5. Program: dropgrp.u (PPL script)

Product: EMS PowerPak 03.00.00.34

Author: Vasu Kaladi
CITI, Georgia Tech
vkaladi@cad.gatech.edu

Usage: ci=dropgrp.u group_name

Comments: Drops the named graphic group from the current file

6. Program: flvigasmems (Unix script)

Author: Vasu Kaladi
CITI, Georgia Tech
e-mail: vkaladi@cad.gatech.edu

Usage: flvigasmems assy_name target_dir

where assy_name is the name of the EMS assembly to be converted (without the extension) which is placed in the directory, target_dir

Comments: This utility is called from "wrigasmems" which places the temporary IGES files for flavoring in target_dir/flavor.

7. Program: leaf.u (PPL script)

Product: EMS PowerPak 03.00.00.34

Author: Vasu Kaladi
CITI, Georgia Tech
vkaladi@cad.gatech.edu

Usage: ci=leaf.u gr_group_name

Comments: This program creates a single named graphic group consisting of all the elements in the current model (a leaf node of the assembly). The name to be given to the graphic group is provided as the first argument to this program

8. Program: parent.u (PPL script)

Product: EMS PowerPak 03.00.00.34

Author: Vasu Kaladi
CITI, Georgia Tech
vkaladi@cad.gatech.edu

Usage: <esc>ci=parent

Comments: Copies elements from the reference files into the current assembly model

9. Program: prochildems (Unix script)

Author: Vasu Kaladi
CITI, Georgia Tech
e-mail: vkaladi@cad.gatech.edu

Usage: prochildems top_assembly_name

Comments: This utility takes as input the name of the top level EMS assembly model file name and calls PPL programs to prepare the EMS assembly model file for output to IGES This is called from the unix script, "wrigasmems".

10. Program: rdchildems (Unix script)

Author: Vasu Kaladi
CITI, Georgia Tech

e-mail: vkaladi@cad.gatech.edu

Usage: rdchildems top_assembly_name

Comments: This utility takes as input the name of the top level IGES assembly model file name and reads in IGES files to create the EMS assembly.
This is called from the unix script, "rdigasmems".

11. Program: rdigasmems (Unix script)

Author: Vasu Kaladi
CITI, Georgia Tech
e-mail: vkaladi@cad.gatech.edu

Usage: rdigasmems top_assembly_name source_dir

Comments: This utility takes as input the name of the top level IGES assembly model file name and the name of the source working directory where the source IGES assembly (from ProEngineer) and all its reference files have been stored.
It then calls the unix utility "rdchildems" to recursively process each node of the assembly tree and creates the EMS assembly model file from IGES.

12. Program: splitc402.scr (IGESWorks script)

Author: Vasu Kaladi
CITI, Georgia Tech
e-mail: vkaladi@cad.gatech.edu

Usage: splitc402.scr sel402 out_dir pfid parent childno

where sel402 is the selection list name for elements in current 402 , out_dir is the output directory, pfid is the file ID of the parent child data file, parent is the name of the parent 402 and childno is the sequence no of the current child of the parent.

Comments: This is called from splitp402.scr.

13. Program: splitp402.scr (IGESWorks script)

Author: Vasu Kaladi
CITI, Georgia Tech

e-mail: vkaladi@cad.gatech.edu

Usage: splitp402.scr in_iges out_dir
where "in_iges" is the input IGES file and "out_dir" is the directory for output IGES files.

Comments: The input IGES files in the current directory that is output from the EMS assembly using "wrigasmems" is split into part IGES files.

14. Program: wrigasmems (Unix script)

Author: Vasu Kaladi
CITI, Georgia Tech
e-mail: vkaladi@cad.gatech.edu

Usage: wrigasmems top_assembly_name target_dir

Comments: This utility takes as input the name of the top level EMS assembly model file name and the name of the target working directory in $HOME/convert/ems where the source EMS assembly and all its reference files have been copied to using "copyasmems". It then calls the unix utility prochildems to recursively process each node of the assembly tree and outputs the EMS assembly model file to IGES After checking inputs, the script changes directory to target_dir for further processing.

15. Program: assemtrmake (Unix script)

Author: Vasu Kaladi
CITI, Georgia Tech
e-mail: vkaladi@cad.gatech.edu

Usage: assemtrmake parent source_dir

Comments: This utility takes as input the name of the parent assembly model file and the name of the source directory where the source EMS assembly and all its reference files are located. It then creates the trail file to assemble the ProE part files that have been created previously from IGES files. This utility is called from "rdigasmproe"

16. Program: copyasmproe (Unix script)

Author: Vasu Kaladi
CITI, Georgia Tech

e-mail: vkaladi@cad.gatech.edu

Usage: copyasmproe top_assembly_name target_dir

Comments: This utility takes as input the name of the top level ProE assembly model file name and the name of the target directory in $HOME/convert/proe to copy the source ProE assembly and all its reference files to.

17. Program: explode.scr (IGESWorks script)

Author: ITI
Modified for PDCF by
Vasu Kaladi
CITI, Georgia Tech
e-mail: vkaladi@cad.gatech.edu

Usage: explode in_iges out_iges

where in_iges is the name of the input IGES file and
out_iges is the name of the output IGES file.

Comments: Flavor Pro/ENGINEER V11.0 Assembly IGES files which contain External File Reference entities (416:1). Entities in the External Reference will be copied to the parent file and referenced from the parent Subfigure Definition.

18. Program: explode416.scr (IGESWorks script)

Author: ITI

Usage: explode416

Comments: Flavor Pro/ENGINEER V11.0 Assembly IGES files which contain External File Reference entities (416:1). Entities in the External Reference will be copied to the parent file and referenced from the parent Subfigure Definition. This is called from the script explode.scr

19. Program: flvigasmproe (Unix script)

Author: Vasu Kaladi
CITI, Georgia Tech
e-mail: vkaladi@cad.gatech.edu

Usage: flvigasmproe assy_name target_dir

where assy_name is the name of the ProE assembly to be converted (without the extension, asm) which is placed in the directory, target_dir

Comments: This utility is called from "wrigasmproe" which places the temporary IGES files for flavoring in target_dir/flavor.

20. Program: getasmstr.scr (IGESWorks script)

Author: Vasu Kaladi
CITI, Georgia Tech
e-mail: vkaladi@cad.gatech.edu

Usage: getasmstr in_iges

where in_iges is the input IGES file

Comments: Extracts the assembly structure (408/308) contained in the input IGES file, in_iges, to "*.child" files.
Uses "getchildstr.scr".

21. Program: getchildstr.scr (IGESWorks script)

Author: Vasu Kaladi
CITI, Georgia Tech
e-mail: vkaladi@cad.gatech.edu

Usage: getchildstr sel308 parent partlist_file

where sel308 is the name of selection list containing elements of the 308 entity, parent is the name of the parent 308 to the current level and partlist_file is the name of the temporary file holding the names of the components of the assembly.

Comments: Called from "getasmstr.scr" and also recursively to extract the assembly structure (408/308) in the IGES file.

22. Program: incr_extn (Unix script)

Author: Vasu Kaladi
CITI, Georgia Tech
e-mail: vkaladi@cad.gatech.edu

Usage: incr_extn file partlist

where "file" is the name to be searched in the data file, "partlist".

Comments: If the data file, "partlist" contains the name, "file", an extension "_n" is added to "file" and appended to "partlist".

23. Program: rdchildproe (Unix script)

Author: Vasu Kaladi
CITI, Georgia Tech
e-mail: vkaladi@cad.gatech.edu
Usage: rdchildproe top_assembly_name

Comments: This utility takes as input the name of the top level assembly and the name of the directory where the source assembly (from EMS) and all its reference files have been stored.
It then creates the ProE assembly.
This is called from the unix script, "rdigasmproe".

24. Program: rdigasmproe (Unix script)

Author: Vasu Kaladi
CITI, Georgia Tech
e-mail: vkaladi@cad.gatech.edu

Usage: rdigasmproe top_assembly_name source_dir

Comments: This utility takes as input the name of the top level assembly and the name of the directory where the source assembly (from EMS) and all its reference files have been stored.
It then calls the unix utility "rdchildproe" to recursively process each node of the assembly tree and creates the ProE assembly model file from IGES.

25. Program: tm408_apply.scr (IGESWorks script)

Author: Vasu Kaladi
CITI, Georgia Tech
e-mail: vkaladi@cad.gatech.edu

Usage: tm408_apply.scr tm408 sel308 out_dir partlist

Comments: This script is called from "tm408fix.scr". The transformation matrix of the 408 element is applied on all the elements in the 308 it references and a separate IGES file is output into "out_dir". The file, "partlist" is used to keep track of IGES file names written out to avoid name conflicts.

26. Program: tm408fix.scr (IGESWorks script)

Author: Vasu Kaladi
CITI, Georgia Tech
e-mail: vkaladi@cad.gatech.edu

Usage: tm408fix.scr in_iges out_dir

Comments: The IGES file output from ProE and flavored using "explode.scr" is split into separate IGES file by applying the transformation matrices of the 408 entities

27. Program: wrigasmproe (Unix script)

Author: Vasu Kaladi
CITI, Georgia Tech
e-mail: vkaladi@cad.gatech.edu

Usage: wrigasmproe assy_name target_dir

where assy_name is the name of the ProE assembly to be converted (without the extension, asm) which is placed in the directory, $HOME/convert/proe/target_dir

Comments: This utility takes as input the name of the top level ProE assembly model file name and the name of the target working directory in $HOME/convert/proe where the source ProE assembly and all its reference files have been copied to. The IGES files are output to $HOME/convert/proe/$target_dir/iges

APPENDIX B

USER GUIDE FOR THE ASSEMBLY CONVERSION PROCESS

In order to illustrate the steps involved in the conversion procedure, consider the sample assembly, *main.asm*. Whenever the name of the sample assembly model is used in the instructions, it is shown in *italics*. The actual name of the model to be converted should be used in its place.

Conversion from ProEngineer to EMS.

1. Log into the pdcf account on the dedicated ProEngineer workstation.

2. Change directory to $HOME/convert/proe

    cd $HOME/convert/proe

3. Make sure that the current directory does not contain a directory or file by the name "assemwrite" in it. Any directory by this name left from a previous conversion should either be deleted or moved to a different name.

3. Run the utility, copyasmproe:

    copyasmproe *main.asm* assemwrite

This copies the *main.asm* assembly and all its component files into the "assemwrite" directory.

4. Run the utility, wrigasmproe

    wrigasmproe *main.asm* assemwrite

This command causes the ProEngineer assembly to be written out as IGES files and then starts a remote shell process on the dedicated IGES/Works workstation to carry out the necessary flavoring actions.

5. Log into the pdcf account on the dedicated EMS workstation.

6. Change directory to $HOME/convert/ems

    cd $HOME/convert/ems

7. Run the utility, "rdigasmems"

rdigasmems *main.asm* $HOME/convert/proe/assemwrite

This command will import the IGES files into EMS design files and also execute the PPL program, "assembl.u" to re-create the assembly in EMS.

The resulting EMS assembly is left in the directory, $HOME/convert/proe/assemwrite/ems.

Conversion from EMS to ProEngineer.

1. Log into the pdcf account on the dedicated EMS workstation.

2. Change directory to $HOME/convert/ems

cd $HOME/convert/ems

3. Make sure that the current directory does not contain a directory or file by the name "assemwrite" in it. Any directory by this name left from a previous conversion should either be deleted or moved to a different name.

3. Run the utility, copyasmems:

copyasmems *main.asm* assemwrite

This copies the *main.asm* assembly and all its component files into the "assemwrite" directory. This utility makes use of EMS' built-in utility for copying assemblies, "revasm" which presents the EMS form titled, "Revise Assembly" on the screen. The user should only "accept and dismiss" this form. This is done by pressing the middle mouse button in the "check box" at the top right hand corner of this form. A message instructing the user to do this is presented before the EMS' "Revise Assembly" form itself is displayed. The user should press the "RETURN" key after reading the instruction message.

4. Run the utility, wrigasmems

wrigasmems *main.asm* assemwrite

This command launches the pre-processing actions in EMS, the IGES export from EMS and also the flavoring actions in IGESWorks through a remote shell process on the dedicated IGES/Works workstation. Parts of the pre-processing actions are carried out in a non-interactive fashion within the EMS environment, while user intervention is required in others. The interactive portion involves copying geometry from the component models into the main assembly. This action takes place under the control of a PPL program. It is therefore imperative

that the instructions that are presented on the screen and the prompt and message areas of the EMS screen be followed carefully. As each component is copied into its parent assembly model, the user is presented with the EMS window displaying only the component to be copied. The user should "select" the geometry displayed by pressing the middle mouse button on the displayed geometry. If all the geometry that is visible is highlighted (except for reference planes), the user should accept it when prompted "Accept or Reject" by pressing the middle mouse in the window. If only a portion of the visible geometry is highlighted, the user should reject it by pressing the right mouse button in the window.

Messages are displayed on the shell window informing the user of the progress of the actions and the user should wait for the Unix prompt to return before proceeding to the next step.

5. Log into the pdcf account on the dedicated ProEngineer workstation.

6. Change directory to $HOME/convert/proe

cd $HOME/convert/proe

7. Run the utility, "rdigasmproe"

rdigasmproe *main.asm* $HOME/convert/ems/assemwrite

This command will import the IGES files into ProEngineer part files and also run a trail file to re-create the assembly in ProEngineer.

The resulting ProEngineer assembly is left in the directory, $HOME/convert/ems/assemwrite/proe.